WHAT IS VULNERABILITY

A vulnerability refers to a weakness in the design or implementation of a system that can be exploited to compromise the security of the system. It is frequently a security loophole that enables an attacker to enter the system by bypassing user authentication..

Common Reasons

Hardware or software misconfiguration

The insecure configuration of the hardware or software in a network can lead to security loopholes. For example, a misconfiguration or the use of an unencrypted protocol may lead to network intrusions, resulting in the leakage of sensitive information.

Insecure or poor design of network and application

An improper and insecure design of a network may make it susceptible to various threats and potential data loss. For example, if firewalls, IDS, and virtual private network (VPN) technologies are not implemented securely, they can expose the network to numerous threats.

Inherent technology weaknesses

If the hardware or software is not capable of defending the network against certain types of attacks, the network will be vulnerable to those attacks. For example, systems running old versions of web browsers are prone to distributed attacks.

End-user carelessness

End-user carelessness considerably impacts network security. Human behavior is fairly susceptible to various types of attacks and can be exploited to effect serious outcomes, including data loss and information leakage.

Intentional end-user acts

Ex-employees who continue to have access to shared drives can misuse them by revealing the company's sensitive information.

Vulnerability Research

- **Microsoft Security Response Center (MSRC)**

Source: https://msrc.microsoft.com

The Microsoft Security Response Center (| MSRC) investigates all reports of security vulnerabilities affecting Microsoft products and services, and it provides information as part of an ongoing effort to help security professionals manage security risks and keep organizational systems protected.

- Packet Storm (https://packetstormsecurity.com)
- Dark Reading (https://www.darkreading.com)
- Trend Micro (https://www.trendmicro.com)
- Security Magazine (https://www.securitymagazine.com)
- PenTest Magazine (https://pentestmag.com)

Vulnerability Assessment

A vulnerability assessment is an in-depth examination of the ability of a system or application, including current security procedures and controls, to withstand exploitation. It scans networks for known security weaknesses, and recognizes, measures, and classifies security vulnerabilities in computer systems, networks, and communication channels. It identifies, quantifies, and ranks possible vulnerabilities to threats in a system. Additionally, it assists security professionals in securing the network by identifying security loopholes or vulnerabilities in the current security mechanism before attackers can exploit them.

There are two approaches to network vulnerability scanning:

Active Scanning: The attacker interacts directly with the target network to find vulnerabilities. Active scanning helps in simulating an attack on the target network to uncover vulnerabilities that can be exploited by the attacker.

Example: An attacker sends probes and specially crafted requests to the target host in the network to identify vulnerabilities.

Passive Scanning: The attacker tries to find vulnerabilities without directly interacting with the target network. The attacker identifies vulnerabilities via information exposed by systems during normal communications. Passive scanning identifies the active operating systems, applications, and ports throughout the target network, monitoring

activity to determine its vulnerabilities.

Example: An attacker guesses the operating system information, applications, and application and service versions by observing the TCP connection setup and teardown.

Limitations of vulnerability assessment

- Vulnerability scanning software is limited in its ability to detect vulnerabilities at a given point in time.
- Vulnerability scanning software must be updated when new vulnerabilities are discovered or when improvements are made to the software being used.
- Software is only as effective as the maintenance performed on it by the software vendor and by the administrator who uses it.
- Vulnerability assessment does not measure the strength of security controls.
- Vulnerability scanning software is not immune to software engineering flaws that might lead to serious vulnerabilities being missed.
- Human judgment is required to analyze the data after scanning and identifying false positives and false negatives.
- Vulnerability scanning software cannot define the impact of an identified vulnerability on different business operations.
- Vulnerability assessment reports are not always easy to understand and assess for risk factors and triage response.
- Vulnerability scanning tools have a narrow focus and do not cover attack vectors such as social engineering.
- Vulnerability scanning software is limited in its ability to perform live tests on web applications to detect errors or unexpected behavior.

Vulnerability scoring systems and databases

Common Vulnerability Scoring System (CVSS)

Source: https://nvd.nist.gov

The CVSS is a published standard that provides an open framework for communicating the characteristics and impacts of IT vulnerabilities. The quantitative model of the system ensures repeatable and accurate measurement while enabling users to view the underlying vulnerability characteristics that were used to generate the scores. Thus, the CVSS is well-suited as a standard measurement system for industries, organizations, and governments that need accurate and consistent vulnerability impact scores.

CVSS assessment consists of the following three metrics for measuring vulnerabilities.

- Base Metric: It represents the inherent qualities of a vulnerability.
- Temporal Metric: It represents the features that continue to change during the lifetime of the vulnerability.
- Environmental Metric: It represents vulnerabilities that are based on a particular environment or implementation.

The metric ranges from 1 to 10, with 10 being the most severe. The CVSS score is calculated and generated by a vector string that represents the numerical score for each group in the form of a block of text.

Vulnerability scoring systems and databases

Common Vulnerabilities and Exposures (CVE)

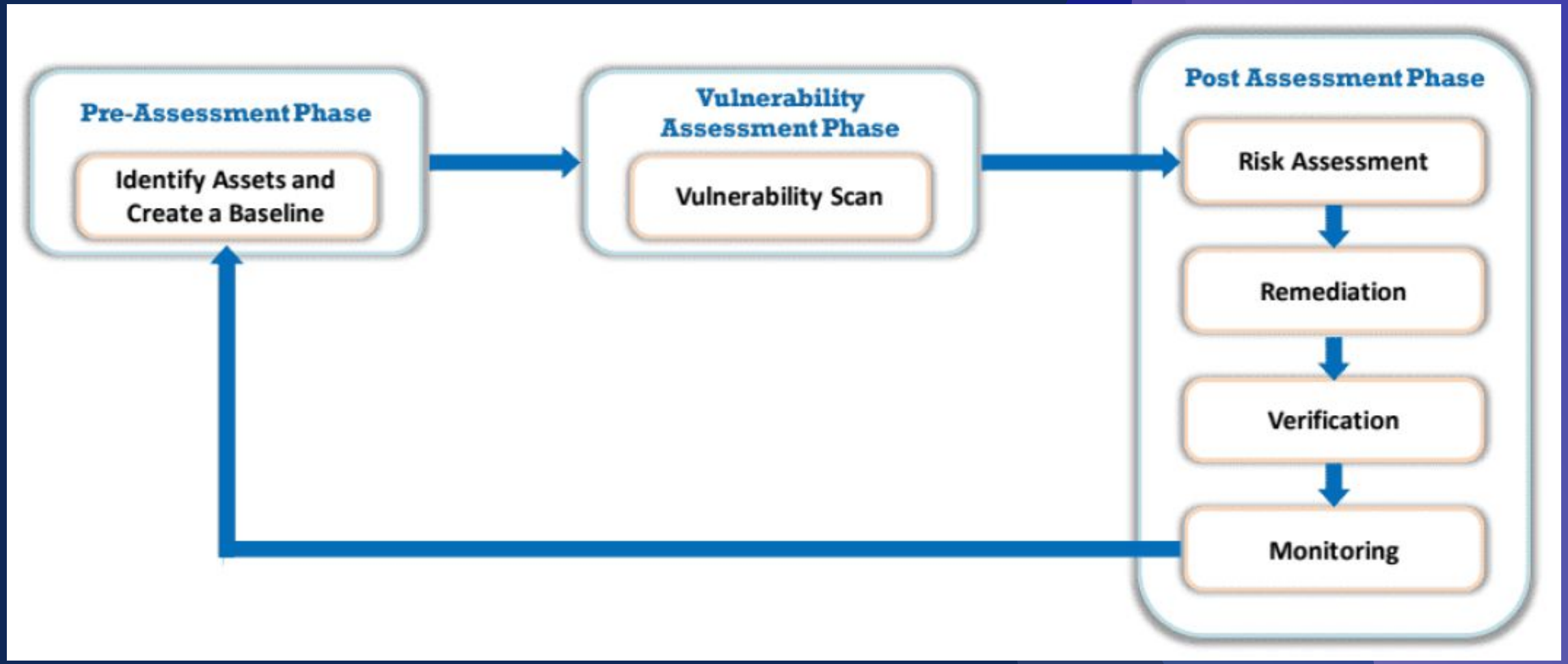
Source: https://www.cve.org

CVE is a publicly available and free-to-use list or dictionary of standardized identifiers for common software vulnerabilities and exposures. The use of CVE Identifiers, or "CVE IDs," which are assigned by CVE Numbering Authorities (CNAs) from around the world, ensures confidence among parties when discussing or sharing information about a unique software or firmware vulnerability. CVE provides a baseline for tool evaluation and enables data exchange for cybersecurity automation.

What CVE is:

- One identifier for one vulnerability or exposure
- One standardized description for each vulnerability or exposure
- A dictionary rather than a database
- A basis for evaluation among services, tools, and databases
- Free for the public to download and use
- Industry-endorsed via the CVE Numbering Authorities, CVE Board, and the numerous products and services that include CVE

Vulnerability Management Life Cycle



Vulnerability Assessment Phase

1. Examine and evaluate the **physical security**
2. Check for **misconfigurations** and human errors
3. Run vulnerability scans
4. Select type of scan based on the organization or **compliance requirements**
5. Identify and **prioritize** vulnerabilities
6. Identify **false positives** and **false negatives**
7. Apply business and technology **context** to scanner results
8. Perform OSINT information gathering to **validate** the vulnerabilities
9. Create a vulnerability scan **report**

Post Assessment Phase

Risk Assessment

- Perform risk categorization
- Assess the level of impact
- Determine the threat and risk levels

Remediation

- Prioritize remediation based on the risk ranking
- Develop an action plan to implement the recommendation/remediation
- Perform root cause analysis
- Apply patches/fixes
- Capture lessons learned
- Conduct awareness training

Verification

- Rescan of systems to identify if applied fix has remediated the vulnerability
- Perform dynamic analysis
- Review of attack surface

Monitoring

- Periodic vulnerability scan and assessment
- Timely remediation of identified vulnerabilities
- Intrusion detection and intrusion prevention logs
- Implementation of policies, procedures, and controls

Classification of vulnerabilities

1. Misconfigurations or Weak Configurations

Setting up a computer network or system in an incorrect manner is a common mistake, mostly because people make errors. These mistakes can make it easy for someone with bad intentions to sneak into the network and access stuff without permission.

2. Network Misconfigurations

- **Open Ports and Services:** Communicating with software over the internet involves open doorways (ports) that should be locked down properly. If these are left open without good security, it can lead to data being stolen or services being shut down by attacks. It's important to check and secure these doorways to keep the network safe.
- **Weak Encryption:** Not scrambling data properly can make it easy for hackers to listen in or change the information being sent around. They can even pretend to be a legitimate service and give false information.
- **Errors:** When applications or services are not set up correctly, they can spill out error messages that give hackers clues on how to break in. Using outdated or flawed software can also give hackers a way to attack remotely.

3. Host Misconfigurations

- **Open Permissions:** Giving users more access to applications or files than necessary can lead to security risks such as private data getting out (data leakage) or system operations damage.
- **Unsecured Root Accounts:** If you stick with the default admin login details that come with your database or software, you could be inviting trouble. Without a strong policy to keep passwords safe, hackers can try different methods to crack these passwords.

Classification of vulnerabilities

4. Application Vulnerabilities

- **Race Conditions:** Occur when system processes or threads vie for a resource and timing affects the outcome, potentially causing software bugs or security issues, like Denial of Service or privilege escalation.
- **DLL Injection:** This happens when a program mistakenly runs a rogue DLL file, possibly leading to the execution of harmful code. Prevention requires using full paths for DLL loading and avoiding untrusted sources.
- **Null Pointer Dereference:** Arises when a program tries to use a null value as a reference, leading to software crashes and potential security breaches. It is often exploited to bypass security and reveal sensitive information.
- **Resource Exhaustion:** This attack overloads a system with excessive requests, similar to a Denial of Service, wasting resources and possibly causing system crashes due to design or coding flaws.
- **Integer Overflows:** When a calculation exceeds the maximum storage capacity for an integer, causing unpredictable behavior, software errors, and potential security gaps like buffer overflows.
- **Buffer Overflows:** Result from coding errors where a program writes data beyond buffer limits, causing crashes or erratic behavior and potentially allowing attackers to execute arbitrary code.
- **Memory Leaks:** Arise when a program doesn't free up memory that's no longer in use, resulting in unnecessary resource consumption and creating opportunities for malicious exploitation.
- **Unsecured Root Accounts:** Using default administrative credentials without a strong password policy can lead to security issues, as attackers might brute-force their way into the system.

Types of vulnerabilities assessment

- **Active Assessment:** Utilizes network scanning to locate hosts, services, and vulnerabilities.
- **Passive Assessment:** Monitors network traffic to identify active components and associated vulnerabilities.
- **External Assessment:** Evaluates the network from an external viewpoint to find exploitable weaknesses.
- **Internal Assessment:** Reviews internal infrastructure for security flaws.
- **Host-based Assessment:** Checks individual system settings to prevent breaches.
- **Network-based Assessment:** Looks for potential network security breaches.
- **Application Assessment:** Scans web infrastructure for misconfigurations or outdated elements.
- **Database Assessment:** Tests databases for vulnerabilities like data exposure or SQL injection.
- **Wireless Network Assessment:** Identifies security issues in wireless networks.
- **Distributed Assessment:** Reviews distributed systems for proper synchronization and security.
- **Credentialed Assessment:** Uses known credentials to assess network security.
- **Non-Credentialed Assessment:** Evaluates network security without login credentials.
- **Manual Assessment:** Ethically hacks into the system to rank and score vulnerabilities.
- **Automated Assessment:** Employs tools to conduct vulnerability assessments.

Approaches to Vulnerability Assessment

1. **Product-Based Solutions:** These are implemented within the organization's internal network. They may be located in a private or non-routable network segment or within Internet-addressable areas. A limitation of product-based solutions is that if they are confined to a private network behind a firewall, they may be unable to detect attacks from outside the network.
2. **Service-Based Solutions:** These are provided by external entities, such as auditing or security consulting organizations. While some service-based solutions are situated within the network, others operate externally. A potential disadvantage of service-based solutions is the possibility that attackers could perform audits on the network from an external vantage point.
3. **Tree-Based Assessment:** In this method, auditors select specific strategies tailored for each type of machine or system component within the IT environment. For example, one type of scanner may be chosen for Windows servers, another for databases, and another for Linux servers. The process depends on the administrator to initiate the scan with some baseline intelligence and proceed to scan continuously without integrating new information during the scan.
4. **Inference-Based Assessment:** This approach begins by cataloging the protocols present on a machine. Once a protocol is identified, the scanning process detects the associated ports and services, such as an email, web server, or database server. Upon identifying services, it targets specific vulnerabilities on each machine and executes tests that are only relevant to the discovered services.

Selection Criteria of Assessment tools

- **Diversity of vulnerabilities detected:** It's crucial to determine the range of vulnerabilities that the tool is capable of identifying.
- **Scanning capabilities:** A competent vulnerability assessment tool should be able to conduct comprehensive tests and scan all the systems that are marked for review.
- **Reporting precision:** Generating concise and clear reports is vital. These reports should offer straightforward strategies for addressing any security flaws found.
- **Scanning efficiency and reliability:** The performance of a scanner is gauged by the speed at which it can assess a single host and the amount of resources it consumes during the scan. Ensuring the results are accurate and that the potential for service interruption is minimized is also critical.
- **Smart Search Ability:** A crucial aspect to consider is the tool's intelligence during the scanning process.
- **Custom Test Creation:** A vulnerability scanning tool gains an advantage if it permits the crafting of custom tests, especially when dealing with new vulnerabilities that lack existing signatures.
- **Scheduling Scans:** The functionality to schedule scans is beneficial, enabling scans to be conducted during times of low network traffic.

Top Vulnerability assessment tools

1. **Nessus Professional:** Nessus Professional is a widely adopted and comprehensive vulnerability scanner that is known for its robust detection capabilities.

2. **OpenVAS:** OpenVAS is an open-source vulnerability scanner and manager that offers a suite of tools for scanning and managing network security.

3. **Nikto:** Nikto is an open-source scanner for web servers that conducts thorough examinations for various items, including more than 6700 files/programs that could pose potential risks.

4. **Qualys Vulnerability Management:** Qualys VM is a cloud service offering immediate, worldwide insight into potential vulnerabilities in IT systems against current digital threats and methods for their protection.

5. **GFI LanGuard:** GFI LanGuard acts as an online security advisor, providing a unified solution that includes patch management, vulnerability scanning, and network audits.

6. **Acunetix:** Acunetix is an automated web vulnerability scanner tool that identifies and reports more than 4500 types of web application vulnerabilities, covering all forms of SQL Injection and XSS.

7. **OWASP ZAP (Zed Attack Proxy) :** OWASP ZAP is an open-source security scanner for web applications designed for users across the spectrum of security expertise. It is perfectly suited for developers, functional testers, and security professionals alike.